```
import hashlib
import random
import re
```

Hashing function

```
def hashingMD5(data_byte):
  return hashlib.md5(data_byte).hexdigest()

def tinyHash(data_byte):
  MD5 = hashingMD5(data_byte)
  return MD5[0:5]
```

Calculate the baseline MD5 for both samplefile.txt and contract.txt

```
sample_file = open('samplefile.txt','rb')
sample_data = sample_file.read()
print("Sample_file MD5: " + hashingMD5(sample_data))
print("Saple_file tinyHash: " + tinyHash(sample_data))
sample_file.close()
```

```
    Sample_file MD5: 5442f94666075ef8d695109af238a5db
    Saple_file tinyHash: 5442f
```

```
contract_file = open('contract.txt','rb')
contract_data = contract_file.read()
print("contract MD5: " + hashingMD5(contract_data))
print("contract tinyHash: " + tinyHash(contract_data))
contract_file.close()
```

```
    contract MD5: f03ac6e0ac6f2b039de8fb4eadd33582
    contract tinyHash: f03ac
```

Generate a random word

```
def randomWord():
  length = random.randint(1,500)
  word = random.randbytes(length)
  return word
```

A function to brute force a collision for Task 1

```
def task1(sample_data):
  tries = 0
  words = randomWord()
  newData = sample_data + randomWord()
  while(tinyHash(newData) != tinyHash(sample_data)):
    words = randomWord()
    newData = sample_data + words
    tries = tries + 1
  return [tries, tinyHash(newData), newData, words]
```

Task 1

```
numberOfFile = 5
result = [[]]*numberOfFile
words = []
fileName = "samplefile.txt"
print(f"Running hash collision for file: {fileName}")
for i in range(numberOfFile):
  result[i] = task1(sample_data)
  words.append(result[i][3])
  #print(words[i])

#print(words)
```

```
    Running hash collision for file: samplefile.txt
```

```
for i in range(numberOfFile):
  number = i+1
  number = i+1
  tries = result[i][0]
  fileName = "collision"+str(number)+".txt"
  print(f"Found TinyHash collision # {number}\t after trying {tries} words.")
  f = open(fileName, "wb")
  f.write(result[i][2])
  f.close()
  print(f"Collision saved as file: {fileName}")
  #print("tinyHash of the file is : " + str(result[i]))
```

```
Found TinyHash collision # 1     after trying 1187979 words.
Collision saved as file: collision1.txt
Found TinyHash collision # 2     after trying 390413 words.
Collision saved as file: collision2.txt
Found TinyHash collision # 3     after trying 2732234 words.
Collision saved as file: collision3.txt
Found TinyHash collision # 4     after trying 236416 words.
Collision saved as file: collision4.txt
Found TinyHash collision # 5     after trying 172818 words.
Collision saved as file: collision5.txt
```

Testing

```
for i in range(numberOfFile):
  collisionFile = "collision" + str(i+1) + ".txt"
  collision = open(collisionFile,'rb')
  collision1 = collision.read()
  print("Collision_file MD5: " + hashingMD5(collision1))
  print("Collision tinyHash: " + tinyHash(collision1))
  collision.close()
```

```
Collision_file MD5: 5442ff2ab96f0e0f58f328d6b4f7c50c
Collision tinyHash: 5442f
Collision_file MD5: 5442f6c326ff56b23e14e2669e87fb53
Collision tinyHash: 5442f
Collision_file MD5: 5442fc16090023a558fa22ed19381d37
Collision tinyHash: 5442f
Collision_file MD5: 5442f3bbe13211bfc49b5f15f3f89a50
Collision tinyHash: 5442f
Collision_file MD5: 5442febcb314f6eabf877d699b0545f7
Collision tinyHash: 5442f
```

A function to brute force Task 2

```
def task2(fileName):
  f = open(fileName, "r")
  data = f.read()
  f.close()
  hash = tinyHash(data.encode())
  #print(hash)
  number = 0
  x = re.search("[$][0-9]*$", data)
  price = int(data[x.start()+1:])
  #print(str(price))
  newData = data[:x.start()+1] + str(number)
  #print(newData)

  while(tinyHash(newData.encode()) != hash):
    number = random.randint(0,price-1)
    newData = data[:x.start()+1] + str(number)
  return [newData, number]
```

Task2

```
print("Starting Task 2...\n")
fileName = "contract.txt"
print(f"Running hash collision for file: {fileName}")
print("Full MD5 digest is: " + str(hashingMD5(contract_data)))
print("TinyHash digest is: " + str(tinyHash(contract_data)))
newNFT = []
newNFT = task2(fileName)
print(f"Found TinyHash collision using this number: {newNFT[1]}")
```

```
Starting Task 2...

Running hash collision for file: contract.txt
```

```
    Full MD5 digest is: f03ac6e0ac6f2b039de8fb4eadd33582
    TinyHash digest is: f03ac
    Found TinyHash collision using this number: 33529
```

Saving the contract to a new file

```
newContract = "newcontract.txt"
f = open(newContract, "w")
f.write(newNFT[0])
f.close
print(f"New contract saved to file: {newContract}")
```

```
    New contract saved to file: newcontract.txt
```

Testing Task2

```
f = open(newContract, "rb")
data = f.read()
print(tinyHash(data))
f.close()
```

```
    f03ac
```

✓ 0s completed at 2:00 PM